University of Saskatchewan
Department of Computer Science

Cmpt 340
Midterm Examination

February 24, 2004

Time: 80 minutes
Total Marks: 73

Professor: A. J. Kusalik
Closed Book

Name: _____

Student Number: _____

Directions:

Answer each of the following questions in the space provided in this exam booklet. If you must continue an answer (e.g. in the extra space on the last page, or on the back side of a page), make sure you clearly indicate that you have done so and where to find the continuation.

Make all written answers legible; no marks can be given for answers which cannot be decrypted. Where a discourse or discussion is called for, please be concise and precise. Do not give "extra answers". Extra answers which are incorrect may result in your being docked marks.

Use of calculators is not allowed during the exam. Fortunately, you should not need a calculator for completing any of the questions.

If you find it necessary to make any assumptions to answer a question, state the assumption with your answer.

Marks for each major question are given at the beginning of that question. There are a total of 72 marks (approximately one mark per minute) .

Good luck.

For marking use only:

A. _____/13	D. _____/3	G. _____/4
B. _____/4	E. _____/9	H. _____/11
C. _____/9	F. _____/5	I. _____/15
		Total: _____/73

A. (maximum 13 marks, scored # right – (# wrong / 2))

For each of the statements below, indicate whether it is **true** ("T") or **false** ("F"). The scoring penalty will not be applied to questions which are not answered (where no response is given).

___ A problem common to procedural languages is the von Neumann bottleneck.

___ One reason for categorizing Prolog as a logic programming language is because its execution mechanism is automated proof construction.

___ In the context of Prolog, another name for an AND-tree is a "search tree".

___ In Prolog, an atom can be any arbitrary string enclosed in single quotes.

___ In Prolog, the anonymous variable is a single occurrence of the hyphen character, "-".

___ In Prolog variable names begin with a lowercase letter, while in Haskell variable names begin with a capital letter.

___ The term

```
np( det( the), noun( cat ), NP )
```

is an example of a ground term.

___ In Prolog, variables in queries are implicitly existentially quantified.

___ In Prolog, a "yes" response to a query can be interpreted as "true", but a response of "no" means "could not be proved".

___ In Haskell, functions always take exactly one argument.

___ Lists and tuples in Haskell correspond to arrays and records, respectively, in procedural languages such as C and Java.

___ In Prolog, an empty binary tree is represented by the atom `void` or `nil`, whereas in Haskell it is terminated by the special symbol ⊥

___ Consider the following data type definition in Haskell:

```
data Tree a = Void | Fork a (Tree a) (Tree a)
```

Here, constructor `Fork` can be considered a function of type

```
a -> Tree a -> Tree a
```

B. (4 marks)

Consider the list `[E,L]` in Prolog. Say whether each of the following is **true** ("T") or **false** ("F"). There will be no scoring penalty applied in this question.

___ The tail of the list is `[L]`.

___ The head of the is is `[E]`.

___ Equivalent shorthand notation is `[E|L]`.

___ Equivalent notation is `.(E,L)`.

C. (3+1+3+1+1 = 9 marks)

Answer each of the following questions with a very short, precise answer.

1. Name three types of programming language paradigms other than procedural or object-oriented.

2. Consider the two Prolog terms `r(st(X))` and `r(Y)`. Give a unifier for these two terms which is <u>not</u> an m.g.u.

3. Recall the following example Prolog program from class

```
student(12345, adelia,     ar,  2002, undeclared).
student(38475, bradford,   ar,  1995, phil).
student(92845, cao,      fgsr,  2001, eng).

class(73463, cmpt, 115, s, t2).
class(47312, cmpt, 317, w, t2).
class(98457, engl, 100, s, t1).
class(16384, phys, 488, w, t2).

enrolled(12345,73463).
enrolled(12345,16384).
enrolled(38475,47312).
enrolled(38475,98457).
enrolled(92845,73463).
enrolled(92845,98457).
```

What Prolog query seeks a solution to the question "Who is a Philosophy major enrolled in a Computer Science in Summer Session?" Give such a query.

4. Consider the following type signatures:

(a) `Int -> [ ( Int -> Int ) ]`

(b) `Int -> [a] -> [a]`

(c) `( Int -> [a]) -> [a]`

(d) `[Int] -> [Int]`

(e) `[Int] -> [a]`

Which of the above type expressions is polymorphic?

5. In the characterization of functional languages given in class, it was stated that a feature of functional languages is the presence of constructs for data abstraction and procedural abstraction. Give the name of a Haskell function defined in the library script *Prelude.hs* of HUGS-98 such that this function provides procedural abstraction.

D. (1+2 = 3 marks)

A student is asked to write a program to implement the predicate `subtree( Subtree, Tree )` which is true if binary tree `Subtree` is a subtree of binary tree `Tree`. It is assumed that trees are formed by functional terms of the form

```
tree( V, L, R )
```

where `V` is the value stored at the root of the tree, `L` is the left subtree, and `R` is the right. An empty tree is represented by the atom `void`.

The student writes the following program:

```
subtree( Tree, Tree ).
subtree( void, Tree ).
subtree( S, tree( _Node, Left, _Right ) ) :-
   subtree( S, Left ).
subtree( S, tree( _Node, _Left, Right ) ) :-
   subtree( S, Right ).
```

Is this program correct? If not, give a query which is in ***M(P)*** but not in ***M***.

Note: the first part of your answer to this question should be a "yes" or a "no" and you should have a supplementary answer only if you specified "no".

E. (2+1+6 = 9 marks)

Consider the relation `last(List,E)` which is true if `List` is a nonempty list and `X` is its last element. The `last/2` predicate can be defined as follows:

```
last( [E], E ).                          % c1
last( [_|Rest], E ) :- last( Rest, E ). % c2
```

1. Give an alternate specification for `last/2` in terms of `append/3`; i.e. rewrite `last/2` using `append/3`.

2. Give a logical consequence of this program. The logical consequence cannot involve the `list` `[a,true,f(3)]`.

3. Give the search tree for the goal

```
?- last([a,true,f(3)], X ).
```

Make sure to show all choice points.

F. (2+1+1+1 = 5 marks)

Consider the function `inc_n` which takes as argument a value `n` and returns the n-th increment function. The latter function, when given a numeric argument, adds `n` to the argument, and returns the result. Function `inc_n` is defined as follows (in Haskell):

```
inc_n n = (+ n)
```

Here is an example of it being used:

```
(inc_n 3) 2
5
```

1. What is the type signature for `inc_n`? **Use no parenthesis.** Recall that `(+)` is defined for all types in type class `Num`.

2. What is the type signature for `inc_n`, this time using parenthesis to explicitly show how the function mapping associates.

3. Is

```
(inc_n 3) 2
```

equivalent to

```
inc_n 3 2
```

? I.e., will they give the same answer? (Yes or No)

(iv) What is the type of `(inc_n 3)` ?

G. (2+2 = 4 marks)

Consider the following Haskell program:

```
f [] [] = True
f (h1:t1) (h2:t2) = if h1==h2 then f t1 t2 else False
```

(a) What does the function above do? I.e., what function does it compute or calculate? Be precise and concise.

(b) The type for `f` above is

```
Ord a => [a] -> [a] -> Bool
```

What would the type signature be of an equivalent function `f1` which performs the same operation, but on a 2-tuple of input values? Give the type signature. I.e. if

```
f1 = uncurry f
```

what is the type of signature of `f1`?

H. (6+5 = 11 marks)

1. Write a function `delneg` in Haskell which takes as argument a list of numbers and returns as its value the same list, but with all negative values removed; e.g.

```
delneg [5, -6, 3, -4, 0] = [5,3,0]
```

The phrase "list of numbers" means that the elements of the list are of some type in the type class `Num`. Make sure to include an explicit type declaration with your function definition.

2. Write a predicate `delneg(List1,List2)` in Prolog which is true if list `List2` is the same list as `List1`, except that all the negative values have been removed. For example

```
?- delneg( [5, -6, 3, -4, 0] , [5,3,0] )
```

is true. Use the built-in Prolog predicates `</2` and `>=/2`. Assume that the arguments will always be lists of integers.

I. (5+4+6 = 15 marks)

Answer each of the following questions with a discussion-oriented answer. Where appropriate, use examples and diagrams to illustrate your points.

1. Give an example of a function or expression which takes advantage of lazy evaluation to return an answer even though part of the computation is undefined or non-terminating. Explain your example, and how lazy evaluation operates in it.

2. What does it mean to say that something is "treated as a first-class object" in the context of modern functional languages? Illustrate with examples.

3. One of Prolog's unique and most powerful features is unification. Depending on the application and usage, unification can be understood as a mechanism for data construction, data transmission, data construction, data extraction, associative searching, activation of processes, message passing, type checking, and pattern-directed procedure invocation.

Explain how unification can be understood as data construction and data extraction. Illustrate with examples.